Gradle Plugins

...
apply **plugin**: **'java'**
apply **plugin**: **'kotlin’**
...

These plugins enable or disable certain feature of languages or frameworks. To use Kotlin in Spring Boot or Spring you will need the following plugins:

- kotlin-allopen
- kotlin-spring
- org.springframework.boot
- io.spring.dependency-management
- kotlin-noarg

Gradle Plugins – **kotlin-allopen**

As you know, Kotlin classes are «**final**» by **default**, which makes it unusable for certain frameworks and libraries such as Spring AOP that requires classes to be «open» . The all-open compiler plugin adapts Kotlin to the requirements of those frameworks and makes annotated classes with a specific annotation and their member variables open **without the explicity «open» keyword**

```
buildscript {
    dependencies {
        classpath "org.jetbrains.kotlin:kotlin-allopen:$kotlin_version»
    }
}

apply plugin: "kotlin-allopen"
```

https://kotlinlang.org/docs/reference/compiler-plugins.html#spring-support

Gradle Plugins – **kotlin-spring**

If you are using Spring, you can enable the **kotlin-spring** compiler plugin instead of specifying Spring annotations manually. The **kotlin-spring is just a wrapper on top of the all-open plugin**, and behaves exactly the same.

(@Component,@Async, @Transactional, @Cachable, @SpringBootTest, @Controller, @RestController, @Service, @Repository)

```
buildscript {
    dependencies {
        classpath "org.jetbrains.kotlin:kotlin-allopen:$kotlin_version»
    }
}

apply plugin: "kotlin-spring"
```

https://kotlinlang.org/docs/reference/compiler-plugins.html#spring-support

Gradle Plugins – org.springframework.boot

The Spring Boot Gradle plugin provides Spring Boot support in Gradle, allowing you to package executable jars or war archives, run Spring Boot applications, and use the dependency management provided by spring-boot-dependencies.

Note it requires Gradle 4

```
dependencies {
    ...
   classpath "org.springframework.boot:spring-boot-gradle-plugin:$spring_boot_version»
    ...
}
apply plugin: 'java' | 'kotlin'
apply plugin: org.springframework.boot
```

https://docs.spring.io/spring-boot/docs/current/gradle-plugin/reference/html

Gradle Plugins – io.spring.dependency-management

When you are apply the io.spring.dependency-managment plugin, Spring Boot's plugin will automatically import the «spring-boot-dependencies» form the version of the Spring Boot that you are using. This provides a similar dependency managment experience to the one that's enjoyed by Maven users ☺

```
apply plugin: 'kotlin'
apply plugin: 'io.spring.dependency-management'

dependencies {
    compile 'org.springframework.boot:spring-boot-starter-web'
    compile 'org.springframework.boot:spring-boot-starter-data-jpa'
}
```

omit the version number

https://docs.spring.io/spring-boot/docs/current/gradle-plugin/reference/html

Gradle - Setup

- Generate a new gradle project in a intellij
- Enable kotlin in the project setup
- Include the needed plugins
  - apply **plugin**: **'kotlin**
    apply **plugin**: **'kotlin-spring'**
    apply **plugin**: **'org.springframework.boot'**
    apply **plugin**: **'io.spring.dependency-management'**
- Add the following modules as dependencies:

```
implementation 'org.springframework.boot:spring-boot-starter-web'
implementation('org.springframework.boot:spring-boot-devtools')
implementation('org.springframework.boot:spring-boot-starter-data-mongodb-reactive')
implementation('org.springframework.boot:spring-boot-starter-webflux')
implementation('org.springframework.boot:spring-boot-starter-security')

implementation('com.fasterxml.jackson.module:jackson-module-kotlin')
implementation('com.fasterxml.jackson.datatype:jackson-datatype-jsr310')
```